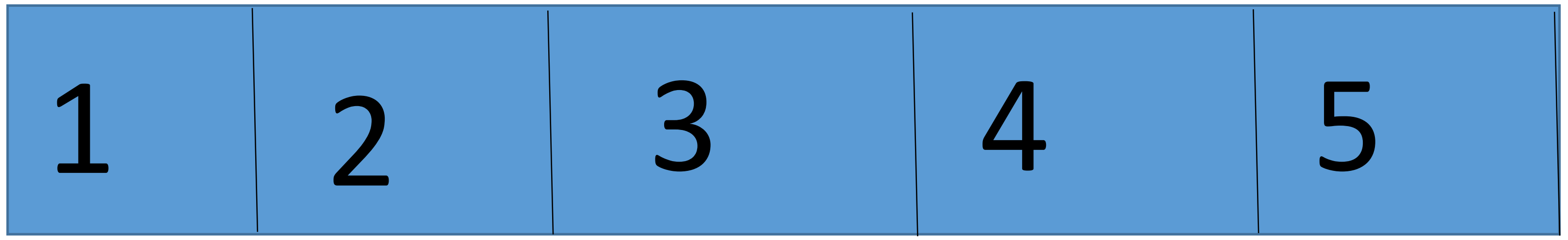
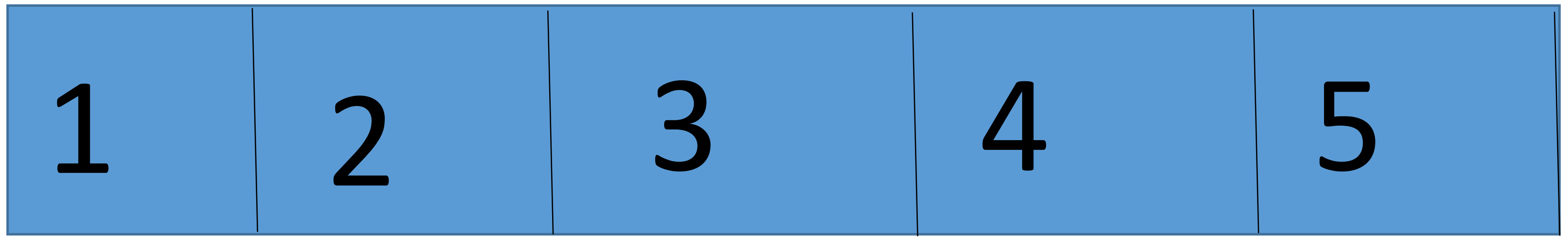
FARIDA - SPOJ



Person cannot pick adjacent cell numbers. So for example if the person picks 1 then he cannot pick 2 and similarly if person picks 3 then he cannot pick 2 and 4.

FARIDA - SPOJ



Option 1
I Should pick 1 here and then go ahead without picking 2.

Option 2
I Should not pick 1 here and then go ahead and try for 2 or others.

FARIDA - SPOJ

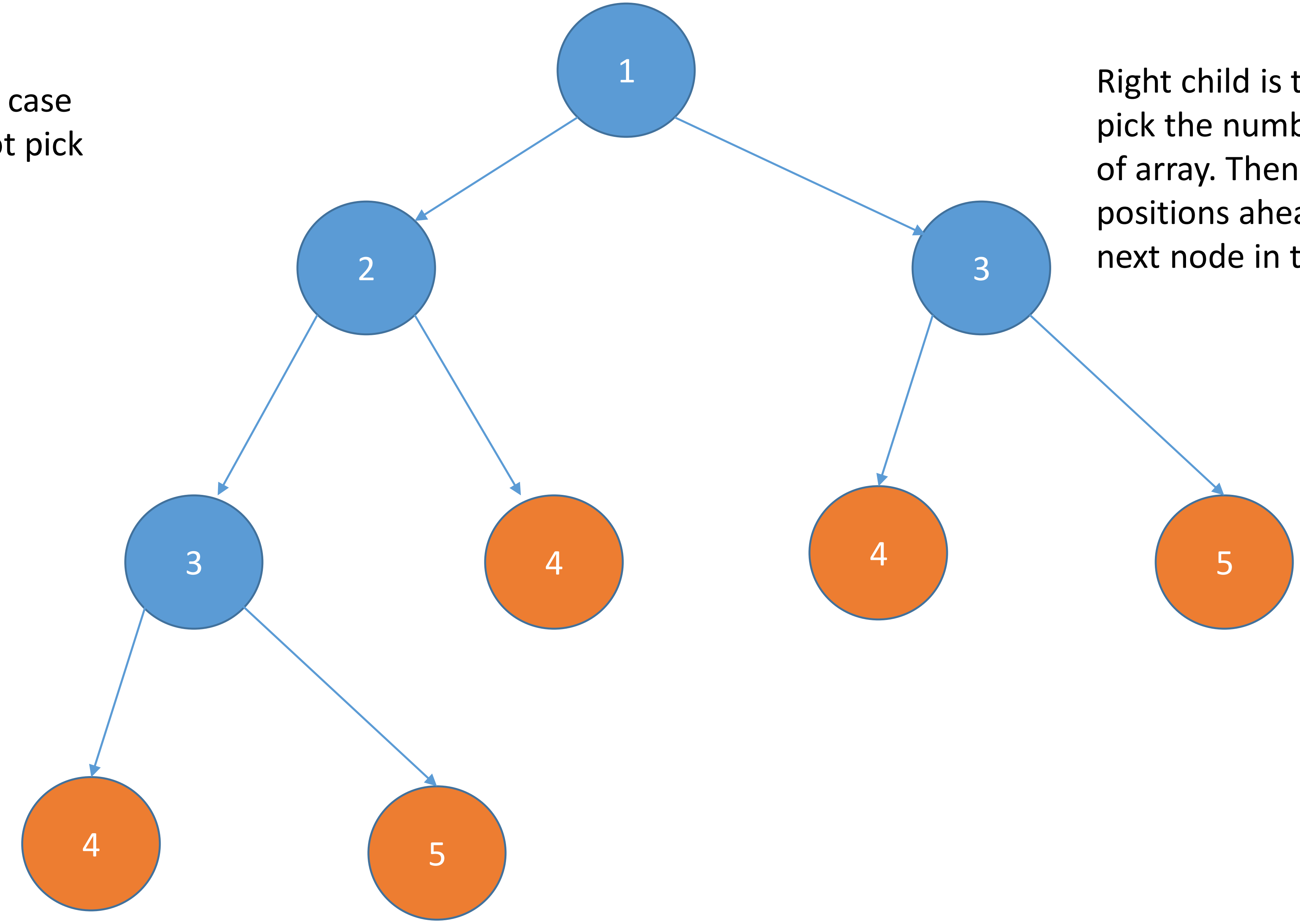
- Since at any cell we have two options and we will try to pick one with maximum answer.

Solve(i) = max(Solve(i+1) , Solve(i+2) + a[i])

Moving Like this guarantees that we will not pick adjacent elements.

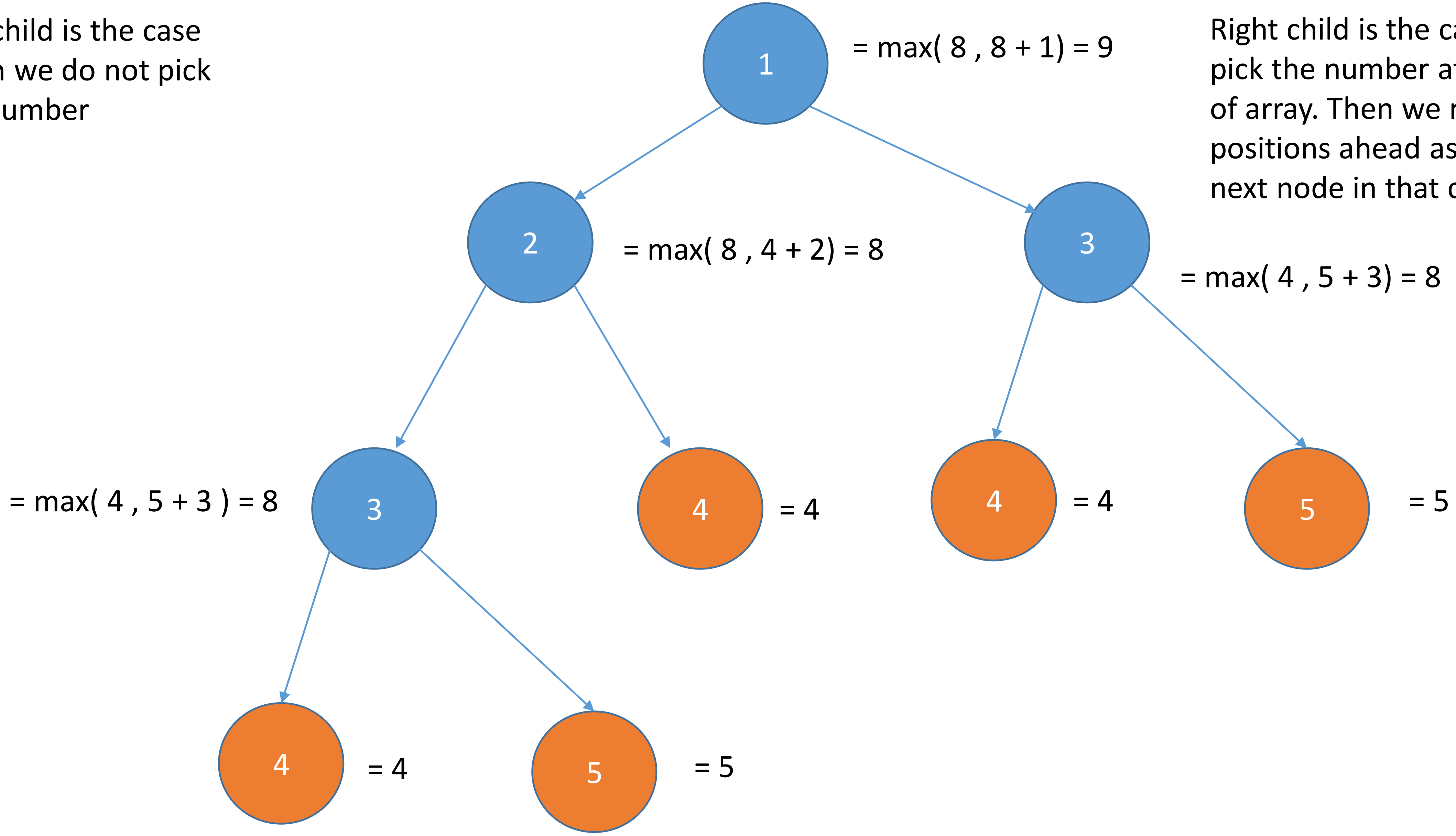
Left child is the case when we do not pick the number

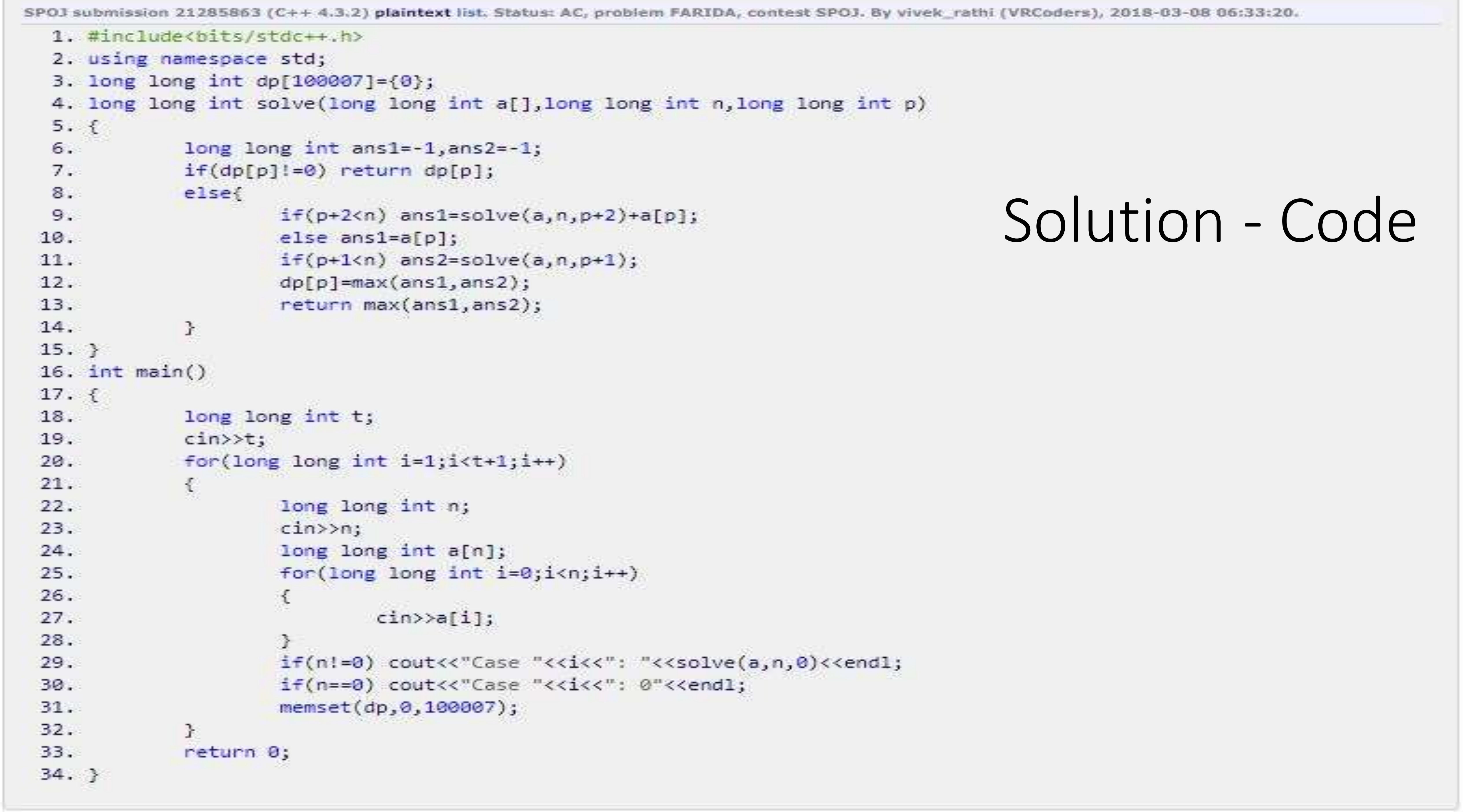
Right child is the case when we pick the number at that position of array. Then we move two positions ahead as we cannot pick next node in that case.



Left child is the case when we do not pick the number

Right child is the case when we pick the number at that position of array. Then we move two positions ahead as we cannot pick next node in that case.





SPOJ submission 21285863 (C++ 4.3.2) plaintext list. Status: AC, problem FARIDA, contest SPOJ. By vivek_rathi (VRCoders), 2018-03-08 06:33:20.

```
#include<bits/stdc++.h>
using namespace std;
long long int dp[100007]={0};
long long int solve(long long int a[],long long int n,long long int p)
{
        long long int ans1=-1,ans2=-1;
        if(dp[p]!=0) return dp[p];
        else{
                if(p+2<n) ans1=solve(a,n,p+2)+a[p];
                else ans1=a[p];
                if(p+1<n) ans2=solve(a,n,p+1);
                dp[p]=max(ans1,ans2);
                return max(ans1,ans2);
        }
}
int main()
{
        long long int t;
        cin>>t;
        for(long long int i=1;i<t+1;i++)
        {
                long long int n;
                cin>>n;
                long long int a[n];
                for(long long int i=0;i<n;i++)
                {
                        cin>>a[i];
                }
                if(n!=0) cout<<"Case "<<i<<": "<<solve(a,n,0)<<endl;
                if(n==0) cout<<"Case "<<i<<": 0"<<endl;
                memset(dp,0,100007);
        }
        return 0;
}
```

Solution - Code